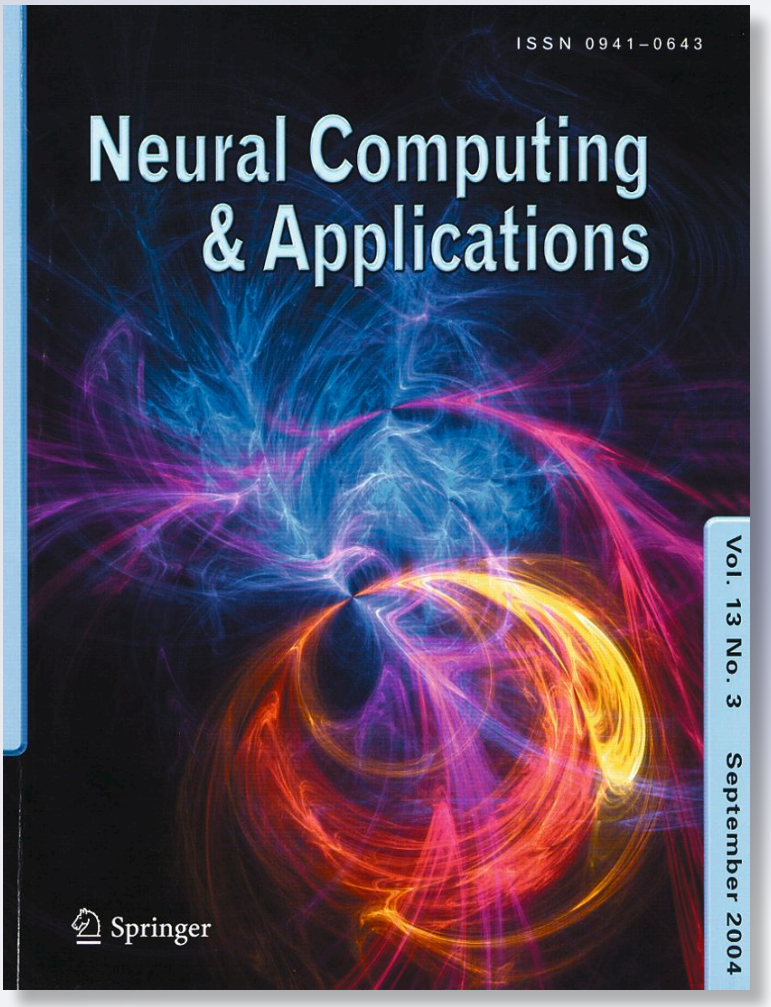
Design of an adaptive self-organizing fuzzy neural network controller for uncertain nonlinear chaotic systems

Chih-Hong Kao, Chun-Fei Hsu & Hon-Son Don

Neural Computing and Applications

ISSN 0941-0643
Volume 21
Number 6

Neural Comput & Applic (2012)
21:1243-1253
DOI 10.1007/s00521-011-0537-2



Springer

Neural Comput & Applic (2012) 21:1243–1253
DOI 10.1007/s00521-011-0537-2

ORIGINAL ARTICLE

Design of an adaptive self-organizing fuzzy neural network controller for uncertain nonlinear chaotic systems

Chih-Hong Kao · Chun-Fei Hsu · Hon-Son Don

Received: 23 August 2010 / Accepted: 28 January 2011 / Published online: 23 February 2011

Abstract Though the control performances of the fuzzy neural network controller are acceptable in many previous published papers, the applications are only parameter learning in which the parameters of fuzzy rules are adjusted but the number of fuzzy rules should be determined by some trials. In this paper, a Takagi–Sugeno-Kang (TSK)-type self-organizing fuzzy neural network (TSK-SOFNN) is studied. The learning algorithm of the proposed TSK-SOFNN not only automatically generates and prunes the fuzzy rules of TSK-SOFNN but also adjusts the parameters of existing fuzzy rules in TSK-SOFNN. Then, an adaptive self-organizing fuzzy neural network controller (ASOFNNC) system composed of a neural controller and a smooth compensator is proposed. The neural controller using the TSK-SOFNN is designed to approximate an ideal controller, and the smooth compensator is designed to dispel the approximation error between the ideal controller and the neural controller. Moreover, a proportional-integral (PI) type parameter tuning mechanism is derived based on the Lyapunov stability theory, thus not only the system stability can be achieved but also the convergence of tracking error can be speeded up. Finally, the proposed ASOFNNC system is applied to a chaotic system. The simulation results verify the system stabilization, favorable tracking performance, and no chattering phenomena can be achieved using the proposed ASOFNNC system.

Keywords Chaotic system · Fuzzy neural network · Neural control · Self-organizing

C.-H. Kao · H.-S. Don
Department of Electrical Engineering,
National Chung-Hsing University,
Taichung 402, Taiwan
e-mail: oakhc888@gmail.com

H.-S. Don
e-mail: honson@ee.nchu.edu.tw

C.-F. Hsu (✉)
Department of Electrical Engineering,
Chung Hua University, Hsinchu 300, Taiwan
e-mail: fei@chu.edu.tw

1 Introduction

If the exact model of the controlled system is well known, there exists an ideal controller to achieve a favorable control performance [1]. A trade-off between the system performance and the model accuracy is necessary for the ideal controller design. The exact models of the nonlinear systems are difficult to develop accurately in real-time applications. To relax this requirement, the neural network-based adaptive controllers have represented an alternative design method for the control of unknown nonlinear systems to compensate the effects of nonlinearities and system uncertainties; so the stability, convergence, and robustness of the control system can be improved [2–6]. Recently, taking the advantages of fuzzy reasoning in handling uncertain information and neural networks in learning from processes, the researches of fuzzy neural networks (FNNs) have attracted the increasing interests [7]. Since the parameterized FNNs can approximate an unknown system dynamics, the FNN-based adaptive control schemes have grown rapidly in many previous published papers [8–12]. The basic issue of the FNN-based adaptive control technique is to provide online learning algorithms that do not require preliminary off-line training.

Though the control performances of the FNN-based adaptive controllers are usually acceptable in [8–12], the

learning algorithm considers only parameter learning in which the parameters of the membership functions and the fuzzy rules are adjusted but the structure of the FNN should be determined in advanced and fixed. For a large number of fuzzy rules, the computation loading is heavy so they are unsuitable for the real-time practical applications. If the number of fuzzy rules is chosen small, the learning performance may be not good enough to achieve a desired control performance due to the inevitable approximation error. Unfortunately, it is difficult to consider the balance between the number of fuzzy rules and the desired performance for the FNN approaches.

To attack the problem of structure determination for FNN, several self-organizing FNN (SOFNN) approaches consist of structure and parameter learning algorithm for FNN, which have been proposed in [13–15]. The self-organizing approach demonstrates the property of automatically generating and pruning fuzzy rules of FNN without the preliminary knowledge. The learning algorithms not only extract the fuzzy rule from input data and adjust the fuzzy partitions of the input and output spaces but also adjust the parameters of existing fuzzy rules. Recently, several SOFNN-based adaptive control schemes have been applied to control the unknown nonlinear systems [16–20]. However, some are too complex; some cannot avoid the structure growing unbounded; and some lack online adaptation ability.

In this paper, a Takagi–Sugeno-Kang (TSK)-type SOFNN (TSK-SOFNN) is studied in which learning algorithm not only automatically generates and prunes the fuzzy rules but also adjusts the parameters of existing fuzzy rules. Then, an adaptive self-organizing fuzzy neural network controller (ASOFNNC) system composed of a neural controller and a smooth compensator is proposed. The neural controller uses the TSK-SOFNN to approximate an ideal controller, and the smooth compensator is utilized to eliminate the approximation error between the neural controller and the ideal controller without occurring chattering phenomena to ensure system stability. Further, this paper derives the proportional-integral (PI) type form adaptation tuning algorithms in the sense of Lyapunov stability to speed up the convergence of the tracking errors and controller parameters. Finally, the proposed ASOFNNC system is applied to a chaotic system. In the simulation study, it is shown that the proposed ASOFNNC system can achieve a favorable tracking performance with rapid convergence of the tracking error and without occurring chattering phenomena. It should be emphasized that the proposed self-organizing method demonstrates the properties of generating and pruning the fuzzy rules automatically with a simple computation.

2 Description of TSK-SOFNN

2.1 Structure learning of TSK-SOFNN

A TSK-SOFNN is shown in Fig. 1, which is comprised of the input, the membership, the rule, and the output layers. Each rule in a TSK-SOFNN is of the following form [7]

$$\text{Rule } i: \text{ IF } q_1 \text{ is } A_1^i \text{ And}\ldots\text{And } q_n \text{ is } A_1^i, \quad \text{THEN } y = \boldsymbol{\alpha}_i^T \mathbf{z} \tag{1}$$

where $\boldsymbol{q} = [q_1, \ldots, q_n]^T$ is the input vector; y is the output variable; $\boldsymbol{\alpha}_i = [\alpha_{i0}, \alpha_{i1}, \ldots, \alpha_{in}]^T$ is the parameter vector designed by the designer; A_j^i is the fuzzy set; and $\mathbf{z} = [1, q_1, \ldots, q_n]^T$. For fuzzy set A_j^i, the Gaussian fuzzy set with membership function is used as

$$\phi_{ij}(q_j) = \exp\left[\frac{-(q_j - c_{ij})^2}{\sigma_{ij}^2}\right] \tag{2}$$

where c_{ij} and σ_{ij} denote the center and width of the fuzzy set A_j^i, respectively. According to the fuzzy AND operation by the algebraic product, the firing strength of the i-th rule is calculated by

$$\Theta_i(\mathbf{q}, \mathbf{c}_i, \boldsymbol{\sigma}_i) = \prod_{j=1}^{n} \phi_{ij}(q_j) \tag{3}$$

where $\mathbf{c}_i = [c_{i1}, \ldots, c_{in}]^T$ and $\boldsymbol{\sigma}_i = [\sigma_{i1}, \ldots, \sigma_{in}]^T$. Assuming there are m rules in the TSK-SOFNN, the output according to the simple weighted sum method would be obtained as

$$y = \sum_{i=1}^{m} \boldsymbol{\alpha}_i^T \mathbf{z} \Theta_i(\mathbf{q}, \mathbf{c}_i, \boldsymbol{\sigma}_i). \tag{4}$$

Then, the output of the TSK-SOFNN represents in a vector form as

$$y = \boldsymbol{\alpha}^T \boldsymbol{\Theta}(\mathbf{q}, \mathbf{c}, \boldsymbol{\sigma}) \tag{5}$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_m^T]^T$; $\boldsymbol{\Theta} = [\Theta_1 \mathbf{q}^T, \ldots, \Theta_m \mathbf{q}^T]^T$; $\mathbf{c} = [\mathbf{c}_1^T, \ldots, \mathbf{c}_m^T]^T$; and $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_1^T, \ldots, \boldsymbol{\sigma}_m^T]^T$.

It is well known that the amount of the fuzzy rules is difficult to select. A trade-off problem between the computation loading and the learning performance arises. This paper proposes that a self-organizing algorithm including how to generate and prune the fuzzy rules of TSK-SOFNN is introduced. The first process of the structure learning is to determine whether to add a new fuzzy rule. If a new input data fall within the boundary of clusters, the TSK-SOFNN will not generate a new fuzzy rule but update parameters of the existing TSK-type fuzzy rules. Consider a distance of mean in association memory as [21]

$$d_i = \|\mathbf{q} - \mathbf{c}_i\|, \quad \text{for} \quad k = 1, 2, \ldots, m. \tag{6}$$

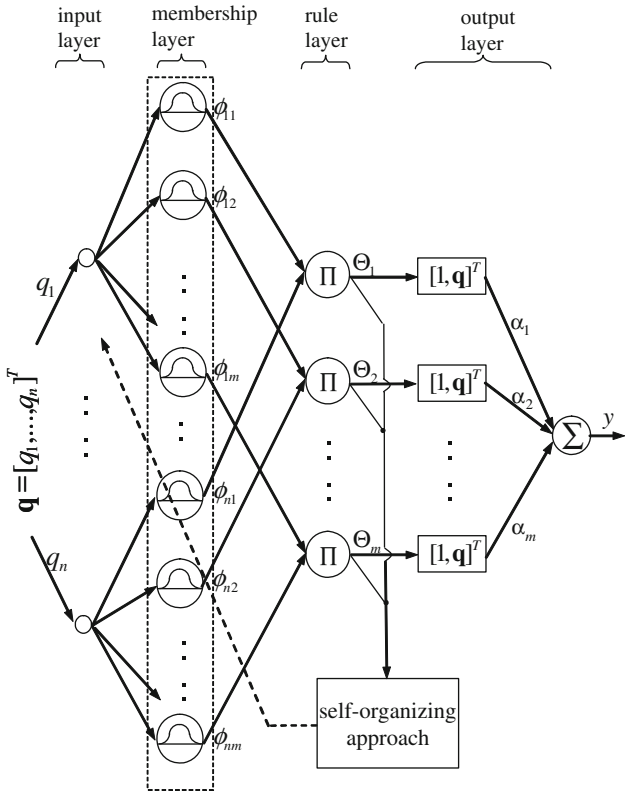


Fig. 1 The architecture of TSK-SOFNN

Find the minimum distance of mean defined as

$$d_{\min} = \min_{1 \le i \le m} d_i. \tag{7}$$

If the distance between input data and the mean is too large for the existing clusters, this means a new cluster should be generated a new input data. It implies if $d_{\min} \ge d_{\text{th}}$ is satisfied, where d_{th} a pre-given threshold, then a new fuzzy rule should be generated. For the new fuzzy rule, the parameters of the new TSK-type fuzzy rule will be defined as

$$\boldsymbol{\alpha}^{\text{new}} = \mathbf{0} \tag{8}$$

$$\mathbf{c}_i^{\text{new}} = \mathbf{q} \tag{9}$$

$$\boldsymbol{\sigma}_i^{\text{new}} = \bar{\boldsymbol{\sigma}} \tag{10}$$

where $\bar{\boldsymbol{\sigma}}$ is a pre-specified vector.

To avoid the endless growing of the TSK-SOFNN structure and the overload computation loading, another self-organizing method is considered to determine whether to delete the existing fuzzy rule but is inappropriate. When the k-th firing strength Θ_k is smaller than a elimination threshold Θ_{th}, it means that the relationship becomes weak between the input and the k-th firing strength. This fuzzy rule may be less or never used. Then, it will gradually reduce the value of the k-th significance index. A significance index determined for the importance of the k-th layer can be given as follows [18]

$$I_k(t+1) = \begin{cases} I_k(t)\exp(-\tau), & \text{if } \Theta_k < \Theta_{\text{th}} \\ I_k(t), & \text{if } \Theta_k \ge \Theta_{\text{th}} \end{cases} \tag{11}$$

where I_k is the significance index of the k-th layer whose initial value is 1 and τ is the elimination speed constant. If $I_k \le I_{\text{th}}$ is satisfied, where I_{th} a pre-given threshold, then the k-th layer will be deleted. The computation loading should be decreased.

2.2 Approximation property of TSK-SOFNN

The main property of TSK-SOFNN regarding feedback control purpose is the universal function approximation property. It implies there exists an expansion of (5) such that it can uniformly approximate a nonlinear function Ω as [13, 18, 22]

$$\Omega = \boldsymbol{\alpha}^{*T}\boldsymbol{\Theta}(\mathbf{q}, \mathbf{c}^*, \boldsymbol{\sigma}^*) + \Delta = \boldsymbol{\alpha}^{*T}\boldsymbol{\Theta}^* + \Delta \tag{12}$$

where Δ is the approximation error; $\boldsymbol{\alpha}^*$ and $\boldsymbol{\Theta}^*$ are the optimal parameter vectors of $\boldsymbol{\alpha}$ and $\boldsymbol{\Theta}$, respectively; and $\mathbf{c}^*$ and $\boldsymbol{\sigma}^*$ are the optimal parameter vectors of $\mathbf{c}$ and $\boldsymbol{\sigma}$, respectively. Since these optimal parameters are unobtainable to best approximation, an estimated TSK-SOFNN is defined as

$$\hat{y} = \hat{\boldsymbol{\alpha}}^T\boldsymbol{\Theta}(\mathbf{q}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}) = \hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\Theta}} \tag{13}$$

where $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\Theta}}$, $\hat{\mathbf{c}}$ and $\hat{\boldsymbol{\sigma}}$ are the estimated values of $\boldsymbol{\alpha}^*$, $\boldsymbol{\Theta}^*$, $\mathbf{c}^*$ and $\boldsymbol{\sigma}^*$, respectively. To speed up the convergence, the optimal parameter vector $\boldsymbol{\alpha}^*$ is decomposed into two parts as [23, 24]

$$\boldsymbol{\alpha}^* = \eta_P\boldsymbol{\alpha}_P^* + \eta_I\boldsymbol{\alpha}_I^* \tag{14}$$

where $\boldsymbol{\alpha}_P^*$ and $\boldsymbol{\alpha}_\mathbf{I}^*$ are the proportional and integral terms of $\boldsymbol{\alpha}^*$, respectively; η_P and η_1 are positive coefficients; and $\boldsymbol{\alpha}_\mathbf{I}^* = \int_0^t \boldsymbol{\alpha}_\mathbf{P}^* \mathrm{d}\tau$. Similarly, the estimation parameter vector $\hat{\boldsymbol{\alpha}}_\mathbf{a}$ is decomposed into two parts as [23, 24]

$$\hat{\boldsymbol{\alpha}} = \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P} + \eta_I\hat{\boldsymbol{\alpha}}_\mathbf{I} \tag{15}$$

where $\hat{\boldsymbol{\alpha}}_\mathbf{P}$ and $\hat{\boldsymbol{\alpha}}_\mathbf{I}$ are the proportional and integral terms of $\hat{\boldsymbol{\alpha}}$, respectively; and $\hat{\boldsymbol{\alpha}}_\mathbf{I} = \int_0^t \hat{\boldsymbol{\alpha}}_\mathbf{P}\mathrm{d}\tau$. Thus, $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}$ can be expressed as

$$\tilde{\boldsymbol{\alpha}} = \eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P} + \eta_P\boldsymbol{\alpha}_\mathbf{P}^* \tag{16}$$

where $\tilde{\boldsymbol{\alpha}}_\mathbf{I} = \boldsymbol{\alpha}_\mathbf{I}^* - \hat{\boldsymbol{\alpha}}_\mathbf{I}$. Define the estimated error $\tilde{y}$ as

$$\begin{aligned} \tilde{y} &= \Omega - \hat{y} \\ &= \boldsymbol{\alpha}^{*T}\boldsymbol{\Theta}^* - \hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\Theta}} + \Delta \\ &= \tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\Theta}} + \Delta \\ &= (\eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P} + \eta_P\boldsymbol{\alpha}_\mathbf{P}^*)^T\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\Theta}} + \Delta \\ &= \eta_I\tilde{\boldsymbol{\alpha}}_I^T\hat{\boldsymbol{\Theta}} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\hat{\boldsymbol{\Theta}} + \eta_P\boldsymbol{\alpha}_\mathbf{P}^{*T}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\Theta}} + \Delta \end{aligned} \tag{17}$$

where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\Theta}} = \boldsymbol{\Theta}^* - \hat{\boldsymbol{\Theta}}$. The Taylor expansion linearization technique is employed to

transform the nonlinear function into a partially linear form [2, 4], i.e.

$$\tilde{\mathbf{\Theta}} = \mathbf{A}^T\tilde{\mathbf{c}} + \mathbf{B}^T\tilde{\boldsymbol{\sigma}} + \mathbf{h} \tag{18}$$

where $\tilde{\mathbf{c}} = \mathbf{c}^* - \hat{\mathbf{c}}$; $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}$; $\mathbf{h}$ is a vector of high order terms; $\mathbf{A} = \left[\frac{\partial\Theta_1}{\partial\mathbf{c}}\frac{\partial\Theta_2}{\partial\mathbf{c}}\cdots\frac{\partial\Theta_m}{\partial\mathbf{c}}\right]|_{c=\hat{c}}$; and $\mathbf{B} = \left[\frac{\partial\Theta_1}{\partial\boldsymbol{\sigma}}\frac{\partial\Theta_2}{\partial\boldsymbol{\sigma}}\cdots\frac{\partial\Theta_m}{\partial\boldsymbol{\sigma}}\right]|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}}$. Substitute (18) into (17), yields

$$\begin{aligned}\tilde{y} &= \eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T\hat{\mathbf{\Theta}} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\hat{\mathbf{\Theta}} + \eta_P\boldsymbol{\alpha}_\mathbf{P}^{*T}\hat{\mathbf{\Theta}} + \hat{\boldsymbol{\alpha}}^T(\mathbf{A}^T\tilde{\mathbf{c}} + \mathbf{B}^T\tilde{\boldsymbol{\sigma}} + \mathbf{h}) \\ &\quad + \tilde{\boldsymbol{\alpha}}^T\tilde{\mathbf{\Theta}} + \Delta \\ &= \eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T\hat{\mathbf{\Theta}} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\hat{\mathbf{\Theta}} + \tilde{\mathbf{c}}^T\mathbf{A}\hat{\alpha} + \tilde{\sigma}^T B\hat{\alpha} + \varepsilon\end{aligned} \tag{19}$$

where $\hat{\boldsymbol{\alpha}}^T\mathbf{A}^T\tilde{\mathbf{c}} = \tilde{\mathbf{c}}^T\mathbf{A}\hat{\alpha}$ and $\hat{\boldsymbol{\alpha}}^T\mathbf{B}^T\tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^T\mathbf{B}\hat{\alpha}$ are used since they are scalars; and $\varepsilon = \hat{\boldsymbol{\alpha}}^T\mathbf{h} + \tilde{\boldsymbol{\alpha}}^T\tilde{\mathbf{\Theta}} + \eta_P\boldsymbol{\alpha}_\mathbf{P}^{*T}\hat{\mathbf{\Theta}} + \Delta$ denotes the lump of approximation error which is assumed to be bounded by $0 \le |\varepsilon| \le E$ in which E is a positive constant.

3 Design of ASOFNNC

3.1 Problem statement

Consider an n-th order class of SISO nonlinear systems described by the following form

$$x^{(n)} = f(\mathbf{x}) + u \tag{20}$$

where $\mathbf{x} = [x, \dot{x}, \ldots, x^{(n-1)}]^T$ is the state vector of the control system which is assumed to be available for measurement; $f(\mathbf{x})$ is the nonlinear system dynamics which can be unknown; and u is the control input. The tracking control problem is to find a control law such the state trajectory x can track a state command x_c closely. Thus, define the tracking error as

$$e = x_c - x. \tag{21}$$

Assume all the parameters in (20) are well known, there exists an ideal controller [1]

$$u^* = -f(\mathbf{x}) + x_c^{(n)} + k_1e^{(n-1)} + \quad \cdots + k_{n-1}\dot{e} + k_ne \tag{22}$$

where k_i, i = 1,2, …, n is positive constant. Applying ideal controller (22) into system dynamic (20), it is obtained

$$\dot{\mathbf{e}} = \mathbf{A_m}\mathbf{e} \tag{23}$$

where $\mathbf{A_m} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ -k_n & -k_{n-1} & \cdots & -k_1 \end{bmatrix}$ and $\mathbf{e} = [e, \dot{e}, \ldots, e^{(n-1)}]^T$ is the state error vector. Suppose the feedback gain k_i is chosen to correspond with the coefficients of a Hurwitz polynomial, it implies that $\lim_{t\to\infty} e = 0$ for any starting initial conditions. Since the system dynamics $f(\mathbf{x})$ may be unknown or perturbed in practical applications, the ideal controller (22) cannot be precisely obtained.

3.2 ASOFNNC system design

To attack this problem for the determination of system dynamics, this paper proposes an ASOFNNC system which is composed of a neural controller and a smooth compensator as shown in Fig. 2, i.e.

$$u_{\text{afnc}} = u_{\text{nc}} + u_{\text{sc}}. \tag{24}$$

The neural controller u_{nc} utilizes the TSK-SOFNN as (13) to mimic the ideal controller in (22), and the smooth compensator u_{sc} is designed to dispel the approximation error introduced by the neural controller in the sense of Lyapunov stability. The self-organizing approach in (6–11) lets the TSK-SOFNN vary its structure dynamically to keep the prescribed approximation accuracy. Substituting (24) into (20) and using (22), the error dynamic equation can be obtained as

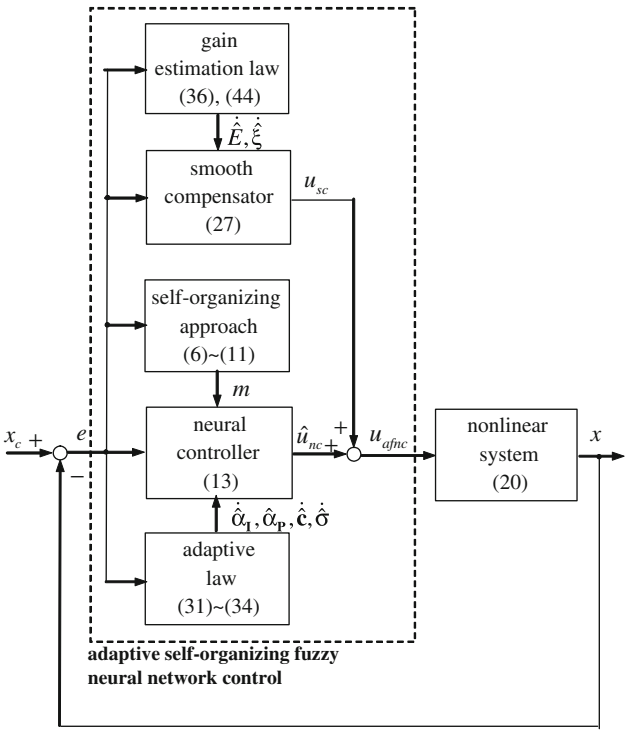


Fig. 2 The block diagram of the ASOFNNC for a class of nonlinear system

$$\dot{\mathbf{e}} = \mathbf{A_m e} + \mathbf{b}(u^* - u_{\rm nc} - u_{\rm sc}) \tag{25}$$

where $\mathbf{b} = [0,0,\ldots 1]^T$. Using the approximation property (19), (25) can be rewritten as

$$\dot{\mathbf{e}} = \mathbf{A_m e} + \mathbf{b}\left(\eta_I \tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^T \hat{\boldsymbol{\Theta}} - \eta_P \hat{\boldsymbol{\alpha}}_{\mathbf{P}}^T \hat{\boldsymbol{\Theta}} + \tilde{\mathbf{c}}^T \mathbf{A}\hat{\alpha} + \tilde{\boldsymbol{\sigma}}^T \mathbf{B}\hat{\alpha} + \varepsilon - u_{\rm sc}\right). \tag{26}$$

Since the number of the fuzzy rules in the TSK-SOFNN is finite for the real-time practical applications, the approximation errors cannot be evitable. To ensure the system stability of the control system, a supervisor compensator was used to dispel the approximation error which requires the bound of the approximation error liking as a sliding-mode controller [8]. If the bound of approximation error chooses too small, it cannot guarantee the system stability in the sense of Lyapunov stability. If the bound of approximation error chooses large to avoid instability, it can be seen that a large bound of approximation error results substantial chattering in the control effort. To cope with this drawback, this paper proposes a smooth compensator as

$$u_{sc} = \begin{cases} \hat{E}\mathrm{sgn}(\mathbf{e}^T\mathbf{Pb}), & \text{for } |\mathbf{e}^T\mathbf{Pb}| > \Phi \\ \hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta} = \hat{E}_P\mathbf{e}^T\mathbf{Pb} + \hat{E}_I\int_0^t (\mathbf{e}^T\mathbf{Pb})d\tau, & \text{for } |\mathbf{e}^T\mathbf{Pb}| \le \Phi \end{cases} \tag{27}$$

where $\hat{\boldsymbol{\zeta}} = [\hat{E}_P, \hat{E}_I]^T$ is a free controller parameter vector; $\boldsymbol{\theta} = [\mathbf{e}^T\mathbf{Pb}, \int(\mathbf{e}^T\mathbf{Pb})\mathrm{d}t]^T$; and Φ is a positive constant which tradeoff between chattering attenuation versus increasing the speed of convergence. When the state trajectory of the system is outside the boundary layer Φ, i.e. $|\mathbf{e}^T\mathbf{Pb}| > \Phi$, the smooth compensator $u_{\rm sc} = \hat{E}\mathrm{sgn}(\mathbf{e}^T\mathbf{Pb})$ is same as a supervisor compensator in [8], and when the state trajectory of the system is inside the boundary layer Φ, i.e. $|\mathbf{e}^T\mathbf{Pb}| \le \Phi$, the smooth compensator $u_{\rm sc} = \hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta}$ is used to eliminate the approximation error between the neural controller and ideal controller. To guarantee the stability of the proposed ASOFNNC system, two cases are considered separately depending on the value of $|\mathbf{e}^T\mathbf{Pb}|$.

For $|\mathbf{e}^T\mathbf{Pb}| > \Phi$, consider the Lyapunov function candidate in the following form as

$$V_1 = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} + \frac{\eta_I}{2}\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^T\tilde{\boldsymbol{\alpha}}_{\mathbf{I}} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}^T\tilde{\mathbf{c}} + \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}} + \frac{1}{2\eta_E}\tilde{E}^2 \tag{28}$$

where the positive constants η_c, η and η_E are the learning rates; $\tilde{E} = E - \hat{E}$ in which $\hat{E}$ is the estimated approximation error bound; and $\mathbf{P}$ is a symmetric positive definite matrix that satisfies the Lyapunov equation

$$\mathbf{A}_{\mathbf{m}}^T\mathbf{P} + \mathbf{PA_m} = -\mathbf{Q} \tag{29}$$

in which $\mathbf{Q}$ sis a positive definite matrix. Taking the derivative of Lyapunov function in (28) and using (26), yields

$$\begin{aligned} \dot{V}_1 &= \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{Pe} + \frac{1}{2}\mathbf{e}^T P\dot{e} + \eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^T\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}} + \frac{\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \frac{\tilde{E}\dot{\tilde{E}}}{\eta_E} \\ &= \frac{1}{2}\mathbf{e}^T\left(\mathbf{A}_{\mathbf{m}}^T\mathbf{P} + \mathbf{PA_m}\right)\mathbf{e} \\ &\quad + \mathbf{e}^T\mathbf{Pb}\left(\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^T\hat{\boldsymbol{\Theta}} - \eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^T\hat{\boldsymbol{\Theta}} + \tilde{\mathbf{c}}^T\mathbf{A}\hat{\alpha} + \tilde{\boldsymbol{\sigma}}^T\mathbf{B}\hat{\alpha} + \varepsilon - u_{\rm sc}\right) \\ &\quad + \eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^T\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}} + \frac{\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \frac{\tilde{E}\dot{\tilde{E}}}{\eta_E} \\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^T\left(\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\Theta}} + \dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}}\right) \\ &\quad - \eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^T\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\Theta}} + \tilde{\mathbf{c}}^T\left(\mathbf{e}^T\mathbf{PbA}\hat{\alpha} + \frac{\dot{\tilde{\mathbf{c}}}}{\eta_c}\right) \\ &\quad + \tilde{\boldsymbol{\sigma}}^T\left(\mathbf{e}^T\mathbf{PB}\hat{\alpha} + \frac{\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma}\right) + \mathbf{e}^T\mathbf{Pb}(\varepsilon - u_{\rm sc}) + \frac{\tilde{E}\dot{\tilde{E}}}{\eta_E} \end{aligned} \tag{30}$$

If the adaptation laws of neural controller choose as

$$\hat{\boldsymbol{\alpha}}_{\mathbf{P}} = \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\Theta}} \tag{31}$$

$$\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}} = -\dot{\hat{\boldsymbol{\alpha}}}_{\mathbf{I}} = -\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\Theta}} \tag{32}$$

$$\dot{\tilde{\mathbf{c}}} = -\dot{\hat{\mathbf{c}}} = -\eta_c\mathbf{e}^T\mathbf{PbA}\hat{\alpha} \tag{33}$$

$$\dot{\tilde{\boldsymbol{\sigma}}} = -\dot{\hat{\boldsymbol{\sigma}}} = -\mathbf{e}^T\mathbf{PB}\hat{\alpha} \tag{34}$$

and the smooth compensator is chosen as

$$u_{\rm sc} = \hat{E}\mathrm{sgn}(\mathbf{e}^T\mathbf{Pb}) \tag{35}$$

with the approximation error bound estimation law

$$\dot{\tilde{E}} = -\dot{\hat{E}} = -\eta_E|\mathbf{e}^T\mathbf{Pb}| \tag{36}$$

then the (30) can be rewritten as

$$\begin{aligned} \dot{V}_1 &= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - \eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^T\hat{\boldsymbol{\alpha}}_{\mathbf{P}} + \varepsilon\mathbf{e}^T\mathbf{Pb} - \hat{E}|\mathbf{e}^T\mathbf{Pb}| \\ &\quad - (E - \hat{E})|\mathbf{e}^T\mathbf{Pb}| \\ &\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + |\varepsilon|\,|\mathbf{e}^T\mathbf{Pb}| - E|\mathbf{e}^T\mathbf{Pb}| \\ &= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - (E - |\varepsilon|)\,|\mathbf{e}^T\mathbf{Pb}| \\ &\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} \le 0. \end{aligned} \tag{37}$$

Since $\dot{V}_1$ is negative semi-definite, that is $V_1(t) \le V_1(0)$, it implies that e, $\tilde{\boldsymbol{\alpha}}$, $\tilde{\mathbf{c}}$, $\tilde{\boldsymbol{\sigma}}$ and $\tilde{E}$ are bounded. Let function $\Xi(t) \equiv \frac{1}{2}\mathbf{e}^T\mathbf{Qe} \le -\dot{V}_1$, and integrate $\Xi(t)$ with respect to time, and it is then obtained as

$$\int_0^t \Xi(\tau)\mathrm{d}\tau \le V_1(0) - V_1(t). \tag{38}$$

Because $V_1(0)$ is bounded, and $V_1(t)$ is nonincreasing and bounded, the following result can be obtained

$$\lim_{t\to\infty} \int_0^t \Xi(\tau)\mathrm{d}\tau < \infty. \tag{39}$$

Also, since $\dot{\Xi}(t)$ is bounded, so by Barbalat's Lemma, it can be shown that $\lim_{t\to\infty} \Xi(t) = 0$. That is $\mathbf{e}(t) \to 0$ as $t \to \infty$ [1]. As a result, the ASOFNNC system with a smooth compensator can be stable for $|\mathbf{e}^T\mathbf{Pb}| > \Phi$.

For $|\mathbf{e}^T\mathbf{Pb}| \le \Phi$, consider the Lyapunov function candidate in the following form as

$$V_2 = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} + \frac{\eta_I}{2}\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T\tilde{\boldsymbol{\alpha}}_\mathbf{I} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}^T\tilde{\mathbf{c}} + \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}} + \frac{1}{2\eta_\xi}\tilde{\boldsymbol{\zeta}}^T\tilde{\boldsymbol{\zeta}} \tag{40}$$

where the positive constant η_ξ is the learning rate; $\tilde{\xi} = \xi^* - \hat{\xi}$ and ξ^* is the optimal value for $\boldsymbol{\xi}$ as defined

$$\zeta^* = \arg\min_{\zeta\in R^2}\left[\sup_{\mathbf{e}^T\mathbf{Pb}\in R}\left|\hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta} - E\mathrm{sgn}(\mathbf{e}^T\mathbf{Pb})\right|\right]. \tag{41}$$

Taking the derivative of Lyapunov function in (40) and using (26), (31–34), yields

$$\begin{aligned}
\dot{V}_2 &= \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{Pe} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + \eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T\dot{\tilde{\boldsymbol{\alpha}}}_\mathbf{I} + \frac{\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi} \\
&= \frac{1}{2}\mathbf{e}^T(\mathbf{A}_\mathbf{m}^T\mathbf{P} + \mathbf{PA}_\mathbf{m})\mathbf{e} + \mathbf{e}^T\mathbf{Pb}(\eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T\hat{\boldsymbol{\Theta}} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\hat{\boldsymbol{\Theta}} + \tilde{\mathbf{c}}^T\mathbf{A}\hat{\alpha} \\
&\quad + \tilde{\boldsymbol{\sigma}}^T\mathbf{B}\hat{\alpha} + \varepsilon - u_{sc}) \quad + \eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T\dot{\tilde{\boldsymbol{\alpha}}}_\mathbf{I} + \frac{\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi} \\
&= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \eta_I\tilde{\boldsymbol{\alpha}}_\mathbf{I}^T(\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\Theta}} + \dot{\tilde{\boldsymbol{\alpha}}}_\mathbf{I}) - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\Theta}} \\
&\quad + \tilde{\mathbf{c}}^T(\mathbf{e}^T\mathbf{PbA}\hat{\alpha} + \frac{\dot{\tilde{\mathbf{c}}}}{\eta_c}) \quad + \tilde{\boldsymbol{\sigma}}^T(\mathbf{e}^T\mathbf{PB}\hat{\alpha} + \frac{\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma}) \\
&\quad + \mathbf{e}^T\mathbf{Pb}(\varepsilon - u_{sc}) + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi} = -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\hat{\boldsymbol{\alpha}}_\mathbf{P} \\
&\quad + \mathbf{e}^T\mathbf{Pb}(\varepsilon - u_{sc}) + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi}
\end{aligned} \tag{42}$$

The smooth compensator is chosen as

$$u_{\mathrm{sc}} = \hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta} \tag{43}$$

with the adaptation law

$$\dot{\tilde{\boldsymbol{\zeta}}} = -\dot{\hat{\boldsymbol{\zeta}}} = -\eta_\xi\mathbf{e}^T\mathbf{Pb}\boldsymbol{\theta} \tag{44}$$

then the (42) can be rewritten as

$$\begin{aligned}
\dot{V}_2 &= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - \eta_P\hat{\boldsymbol{\alpha}}_\mathbf{P}^T\hat{\boldsymbol{\alpha}}_\mathbf{P} + \mathbf{e}^T\mathbf{Pb}(\varepsilon - \hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta}) + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi} \\
&\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \mathbf{e}^T\mathbf{Pb}(\varepsilon - \hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta}) + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi} \\
&= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \mathbf{e}^T\mathbf{Pb}(\varepsilon + \tilde{\boldsymbol{\zeta}}^T\boldsymbol{\theta} - \boldsymbol{\zeta}^{*T}\boldsymbol{\theta}) + \frac{\tilde{\boldsymbol{\zeta}}^T\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi} \\
&= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \varepsilon\mathbf{e}^T\mathbf{Pb} + \tilde{\boldsymbol{\zeta}}^T(\mathbf{e}^T\mathbf{Pb}\theta + \frac{\dot{\tilde{\boldsymbol{\zeta}}}}{\eta_\xi}) - \mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta} \\
&= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \varepsilon\mathbf{e}^T\mathbf{Pb} - \mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta} \\
&\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + |\varepsilon||\mathbf{e}^T\mathbf{Pb}| - \mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta} \\
&\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + E|\mathbf{e}^T\mathbf{Pb}| - \mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta}.
\end{aligned} \tag{45}$$

From the definition of (27), it can find $\mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta}$ lies in the first and third quadrant. So $\mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta} = 0$ for $\mathbf{e}^T\mathbf{Pb} = 0$ and $\mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta} \ge 0$ for all $\mathbf{e}^T\mathbf{Pb}$. It can find $\mathbf{e}^T\mathbf{Pb}\boldsymbol{\zeta}^{*T}\boldsymbol{\theta} = |\mathbf{e}^T\mathbf{Pb}||\boldsymbol{\zeta}^{*T}\boldsymbol{\theta}|$, thus (45) can be rewritten as

$$\begin{aligned}
\dot{V}_2 &\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + E|\mathbf{e}^T\mathbf{Pb}| - |\boldsymbol{\zeta}^{*T}\boldsymbol{\theta}||\mathbf{e}^T\mathbf{Pb}| \\
&= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - (|\boldsymbol{\zeta}^{*T}\boldsymbol{\theta}| - E)|\mathbf{e}^T\mathbf{Pb}| \\
&\le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} \le 0.
\end{aligned} \tag{46}$$

Similar to the proof of (37), it can be similarly shown $\mathbf{e}(t) \to 0$ as $t \to \infty$. As a result, the ASOFNNC system with a smooth compensator can be stable for $|\mathbf{e}^T\mathbf{Pb}| \le \Phi$.

4 Simulation results

Chaotic systems have been studied and known to exhibit complex dynamical behavior. The interest in chaotic systems lies mostly upon their complex, unpredictable behavior, and extreme sensitivity to initial conditions as well as parameter variations. The issue of the chaotic controller design has become a significant research topic in the physics, mathematics, and engineering communities [25–29]. This study considers a second-order chaotic system as follow [25]

$$\ddot{x} = -p\dot{x} - p_1x - p_2x^3 + q\cos(\omega t) + u = f(\mathbf{x}) + u \tag{47}$$

where $x = [x, \dot{x}]^T$ is the state vector of the system; $f(\mathbf{x}) = -p\dot{x} - p_1x - p_2x^3 + q\cos(\omega t)$ is the system dynamic function; u is the control effort; and p, p_1, p_2, q and ω are real constants. For observing these complex phenomena, the open-loop chaotic system behavior with $u = 0$ was

simulated with $p = 0.4$, $p_1 = 0.4$, $p_1 = -1.1$, $p_2 = 1.0$ and $\omega = 1.8$. For the phase plane plots from an initial condition point (0, 0), an uncontrolled trajectory of chaotic with $q = 2.1$ and $q = 7.0$ are shown in Fig. 3a, b, respectively. It is shown that the uncontrolled chaotic system has different chaotic trajectories with different system parameters [25]. To illustrate the effectiveness of the proposed ASOFNNC system, a comparison among FNN-based adaptive controller in [8], ASOFNNC system with an integral type adaptation law, and ASOFNNC system with a PI type adaptation law is made.

First, the FNN-based adaptive controller in [8] is applied to the chaotic system. The simulation results of the FNN-based adaptive controller with 5 fuzzy rules are shown in Figs. 4 and 5 for $q = 2.1$ and $q = 7.0$, respectively. The structure of the used FNN was determined by some trial. The tracking responses of state x are shown in Figs. 4a and 5a; the tracking responses of state $\dot{x}$ are shown in Figs. 4b and 5b; and the associated control efforts are shown in

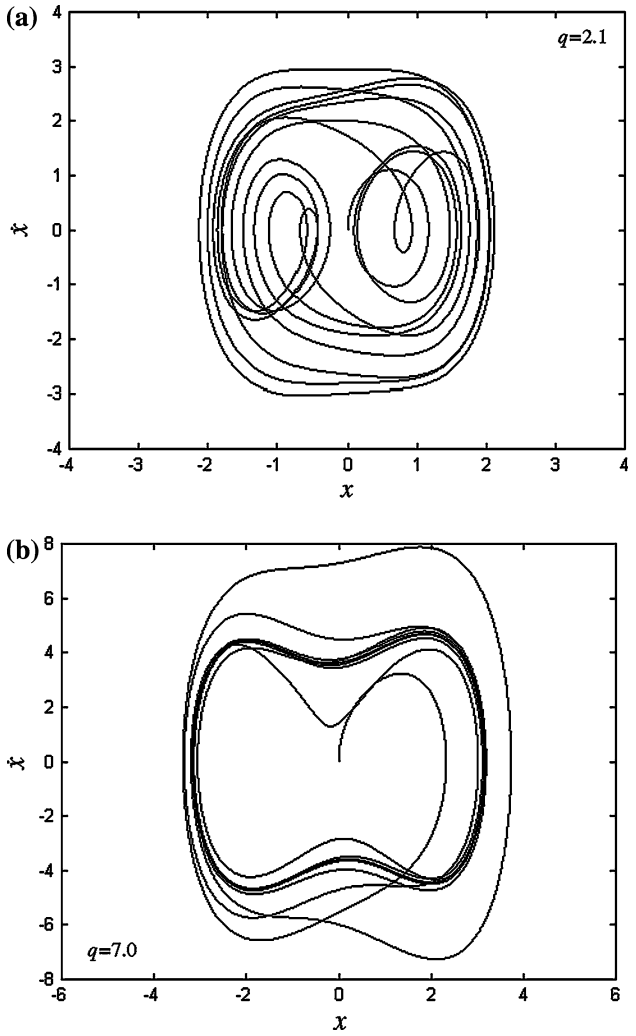


Fig. 3 The uncontrolled Duffing's chaotic system

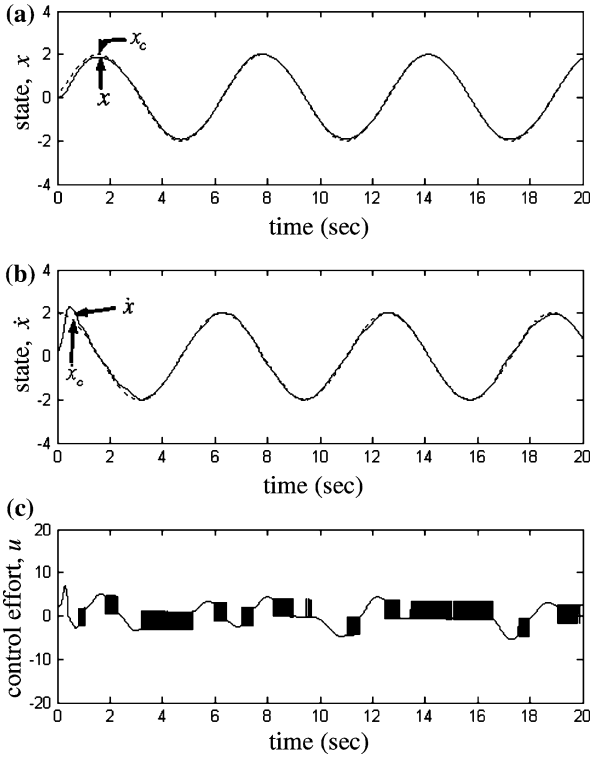


Fig. 4 Simulation results of the FNN-based adaptive controller for $q = 2.1$

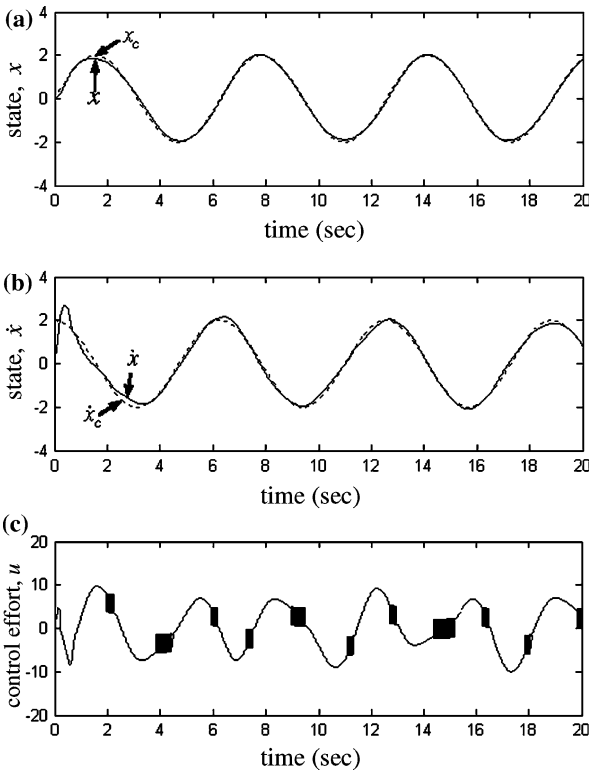


Fig. 5 Simulation results of the FNN-based adaptive controller for $q = 7.0$

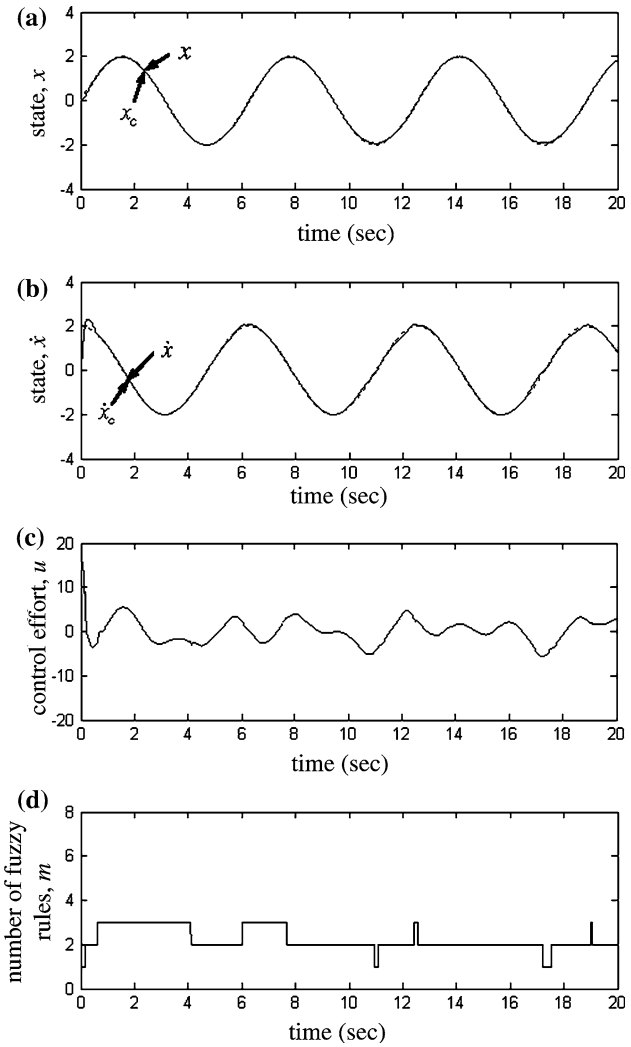


Fig. 6 Simulation results of the ASOFNNC system with an integral type adaptation law for $q = 2.1$

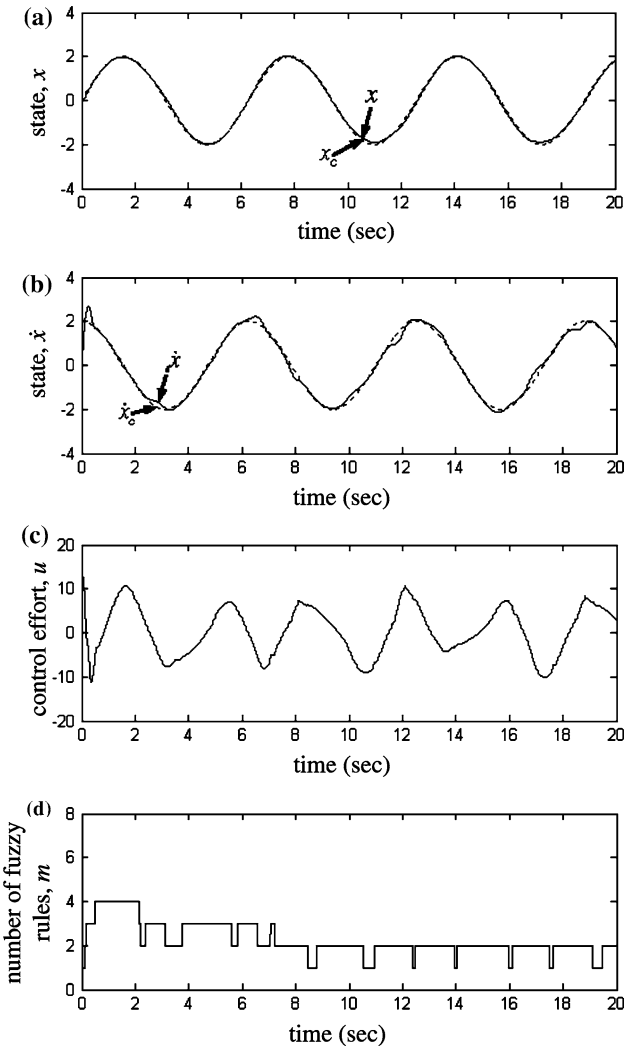


Fig. 7 Simulation results of the ASOFNNC system with an integral type adaptation law for $q = 7.0$

Figs. 4c and 5c. The simulation results show that a robust tracking performance can be achieved after the controller parameters being well learned. Unfortunately, to guarantee the system stability, a switching compensator should be used, but the undesirable chattering phenomenon occurs as shown in Figs. 4c and 5c.

Then, the proposed ASOFNNC system is applied to the chaotic system again. It should be emphasized that the development of the ASOFNNC scheme does not need to know the system dynamics. For the practical implementation, the controller parameters of the ASOFNNC system can be tuned online by the developed adaptive laws. For a choice of $\mathbf{Q} = \mathbf{I}$, solve the Riccati-like Eq. (29), then

$$\mathbf{P} = \begin{bmatrix} 3.2250 & 3.1250 \\ 3.1250 & 4.5312 \end{bmatrix}. \tag{48}$$

The control parameters of the proposed ASOFNNC system are chosen as k_1 =0.8, k_2 =0.16, $\eta_P = \eta_I = 10$, $\eta_c = \eta_\sigma = 1$, $\eta_E = \eta_\varepsilon = 0.1$, $d_{\text{th}} = 0.6$, $\bar{\boldsymbol{\sigma}} = 0.3$, $\Theta_{th} = 0.1$, $\tau = 0.01$ and $I_{\text{th}} = 0.01$. The choice of these parameters is also through some trails, and all the gains are chosen in consideration of the requirement of stability. To compare the convergence speed of the tracking error, the ASOFNNC system with an integral type learning algorithm is applied first. This is a special case of the proposed ASOFNNC system with a PI type learning algorithm for $\eta_P = 0$. The simulation results of the ASOFNN system with an integral type learning algorithm are shown in Figs. 6 and 7 for $q = 2.1$ and $q = 7.0$, respectively. The tracking responses of state x are shown in Figs. 6a and 7a; the tracking responses of state $\dot{x}$ are shown in Figs. 6b and 7b; the associated control efforts are shown in Figs. 6c and 7c; and the numbers of fuzzy rules are shown in Figs. 6d and 7d. The simulation results show that the proposed ASOFNNC system with an integral type learning algorithm not only can achieve a favorable tracking performance but also an

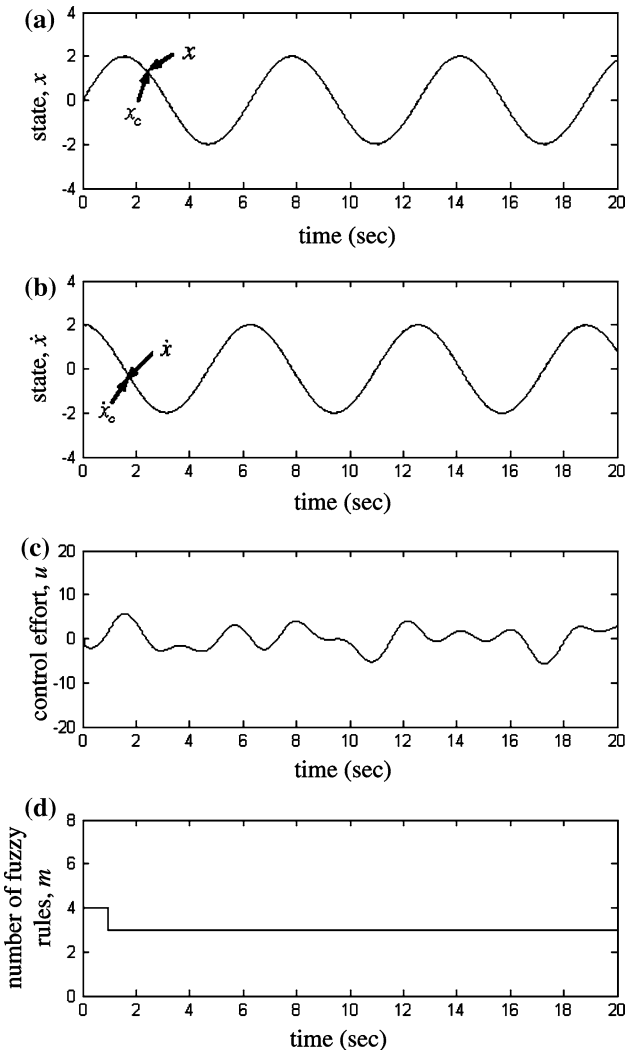


Fig. 8 Simulation results of the ASOFNNC system with a PI type adaptation law for $q = 2.1$

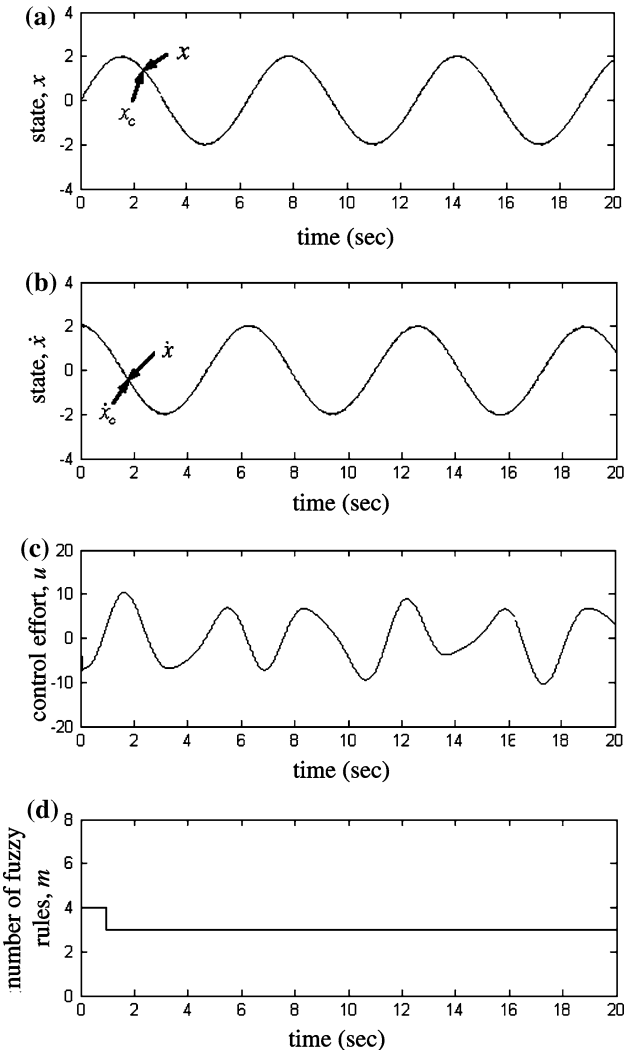


Fig. 9 Simulation results of the ASOFNNC system with a PI type adaptation law for $q = 7.0$

appropriate network size can be obtained since the proposed self-organizing mechanism is applied. Since the smooth compensator is designed as $\hat{E}\text{sgn}(\mathbf{e}^T\mathbf{Pb})$ outside the boundary layer and is designed as $\hat{\boldsymbol{\zeta}}^T\boldsymbol{\theta}$ inside the boundary layer to attenuate the effects of the approximation errors, there are no chattering phenomena in Figs. 6c and 7c. However, the convergence speed of the tracking error is slow using an integral type learning algorithm.

Finally, the PI type learning algorithm is applied with $\eta_P = 10$. The simulation results of the ASOFNN system with a PI type learning algorithm are shown in Figs. 8 and 9 for $q = 2.1$ and $q = 7.0$, respectively. The tracking responses of state x are shown in Figs. 8a and 9a; the tracking responses of state $\dot{x}$ are shown in Figs. 8b and 9b; the associated control efforts are shown in Figs. 8c and 9c; and the numbers of fuzzy rules are shown in Figs. 8d and 9d. The simulation results show that the proposed ASOFNNC system with a PI type learning algorithm can achieve a favorable tracking performance if the controller parameters are well trained. The used TSK-SOFNN varies its structure dynamically to keep the prescribed approximation accuracy with a simple computation. Moreover, it does not cause the chattering phenomena in the associated control efforts, and the convergence speed of the tracking error is accelerated by the PI type learning algorithm.

For further performance, comparison among the aforementioned control schemes, a performance index $I = \sum_t e^2 + \dot{e}^2$ is considered. The performance indices of FNN-based adaptive controller, ASOFNNC system with an integral type adaptation law, and ASOFNNC system with a PI type adaptation law are shown Fig. 10a, b for $q = 2.1$ and $q = 7.0$, respectively. It is shown that the performance index of the proposed ASOFNNC system with a PI type adaptation law is smaller than those of the other methods.

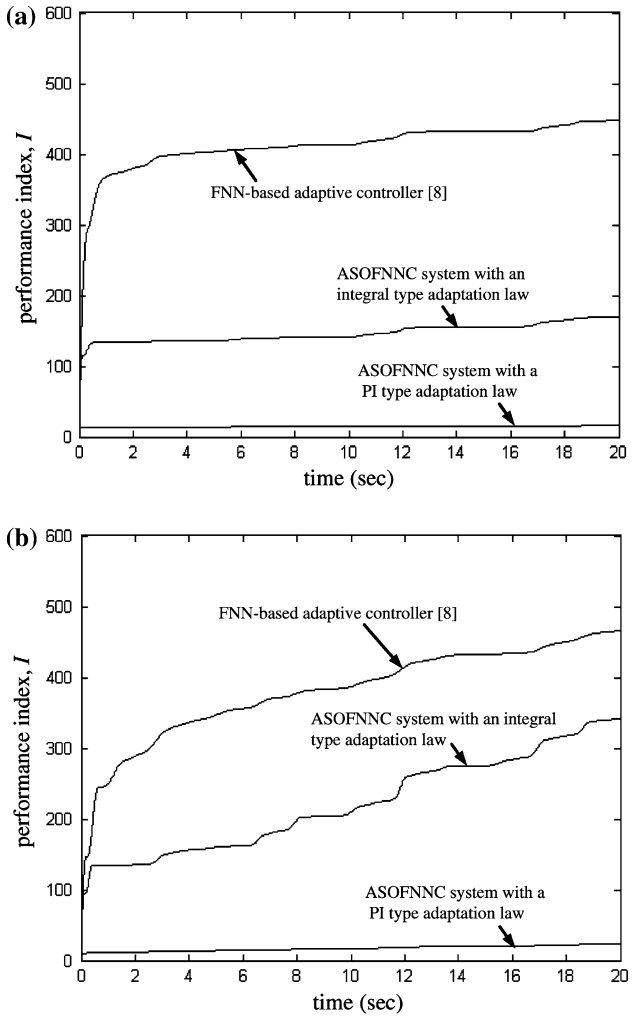


Fig. 10 Comparison of performance indices

This is due to the fact that the tracking errors converge the most quickly by using the proposed ASOFNNC system with a PI type adaptation law.

5 Conclusions

In this paper, an adaptive self-organizing fuzzy neural network controller (ASOFNNC) system has been successfully applied to a chaotic system. All the controller parameters of the proposed ASOFNNC system online tune in the sense of Lyapunov stability; thus, the system stability can be guaranteed. A comparison of control characteristics among FNN-based adaptive controller in [8], ASOFNNC system with an integral type adaptation law, and ASOFNNC system with a PI type adaptation law is summarized in Table 1. It is shown that the ASOFNNC system with a PI type adaptation law has the fast transient response and without occurring chattering phenomena to ensure system stability.

Table 1 Characteristic comparison

Controller	Controller parameters	Network structure	Convergence speed	Chattering phenomenon
FNN-based adaptive control [8]	Online learning	Trial and error	Middle	Serious
ASOFNNC with an integral type adaptation law	Online learning	Online learning	Slow	None
ASOFNNC with a PI type adaptation law	Online learning	Online learning	Fast	None

In summary, the major contributions of this paper are as follows: (1) the developed TSK-type self-organizing fuzzy neural network varies its structure dynamically to keep the prescribed approximation accuracy with a simple computation, (2) the successful development of the ASOFNNC scheme in the sense of Lyapunov stability, (3) the proportional-integral type learning algorithm is designed to achieve a better tracking performance, (4) the smooth compensator can guarantee system stability without occurring chattering phenomena, and (5) the successful applications of the ASOFNNC system to a chaotic system.

Acknowledgments The authors appreciate partial support from the National Science Council of Republic of China under grant NSC 98-2221-E-216-040. The authors would like to express their gratitude to the reviewers for their valuable comments and suggestions.

References

1. Slotine JJE, Li WP (1991) Applied nonlinear control. Prentice-Hall, Englewood Cliffs
2. Lin CM, Peng YF (2005) Missile guidance law design using adaptive cerebellar model articulation controller. IEEE Trans Neural Netw 16(3):636–644
3. Duarte-Mermoud MA, Suarez AM, Bassi DF (2005) Multivariable predictive control of a pressurized tank using neural networks. Neural Comput Appl 15(1):18–25
4. Hsu CF, Lin CM, Lee TT (2006) Wavelet adaptive backstepping control for a class of nonlinear systems. IEEE Trans Neural Netw 17(5):1175–1183
5. Wang Z, Zhang Y, Fang H (2008) Neural adaptive control for a class of nonlinear systems with unknown deadzone. Neural Comput Appl 17(4):339–345
6. Hsu CF (2009) Design of intelligent power controller for DC-DC converters using CMAC neural network. Neural Comput Appl 18(1):93–103
7. Lin CT, Lee CSG (1996) Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. Prentice-Hall, Englewood Cliffs
8. Lin CM, Hsu CF (2004) Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings. IEEE Trans Fuzzy Syst 12(5):733–742

9. Leu YG, Wang WY, Lee TT (2005) Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems. IEEE Trans Neural Netw 16(4):853–861
10. Cheng KH, Hsu CF, Lin CM, Lee TT, Li C (2007) Fuzzy-neural sliding-mode control for DC-DC converters using asymmetric Gaussian membership functions. IEEE Trans Ind Electron 54(3):1528–1536
11. Da F (2007) Fuzzy neural network sliding mode control for long delay time systems based on fuzzy prediction. Neural Comput Appl 17(5):531–539
12. Chen CS, Chen HH (2009) Robust adaptive neural-fuzzy-network control for the synchronization of uncertain chaotic systems. Nonlinear Anal Real World Appl 10(3):1466–1479
13. Juang CF, Lin CT (1998) An on-line self-constructing neural fuzzy inference network and its applications. IEEE Trans Fuzzy Syst 6(1):12–32
14. Lin CT, Cheng WC, Liang SF (2005) An on-line ICA-mixture-model-based self-constructing fuzzy neural network. IEEE Trans Circuits Syst I 52(1):207–221
15. Juang CF, Wang CY (2009) A self-generating fuzzy system with ant and particle swarm cooperative optimization. Expert Syst with Appl 36(3):5362–5370
16. Gao Y, Er MJ (2003) Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems. IEEE Trans Fuzzy Syst 11(4):462–477
17. Lin FJ, Lin CH (2004) A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller. IEEE Trans Energy Conversion 19(1):66–72
18. Hsu CF (2007) Self-organizing adaptive fuzzy neural control for a class of nonlinear systems. IEEE Trans Neural Netw 18(4):1232–1241
19. Lin D, Wang X (2010) Observer-based decentralized fuzzy neural sliding mode control for interconnected unknown chaotic systems via network structure adaptation. Fuzzy Sets Syst 161(15):2066–2080
20. Cheng KH (2009) Auto-structuring fuzzy neural system for intelligent control. J Franklin Inst 346(3):267–288
21. Lin CM, Chen TY (2009) Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems. IEEE Trans Neural Netw 20(9):1377–1384
22. Wang LX (1994) Adaptive fuzzy systems and control: design and stability analysis. Prentice-Hall, Englewood Cliffs
23. Golea N, Golea A, Benmahammed K (2002) Fuzzy model reference adaptive control. IEEE Trans Fuzzy Syst 10(4):436–444
24. Hsu CF, Chung CM, Lin CM, Hsu CY (2009) Adaptive CMAC neural control of chaotic systems with a PI-type learning algorithm. Expert Syst with Appl 36(9):11836–11843
25. Chen G, Dong X (1993) On feedback control of chaotic continuous time systems. IEEE Trans Circuits Syst I 40(9):591–601
26. Chen HK (2002) Chaos and chaos synchronization of a symmetric gyro with linear-plus-cubic damping. J Sound Vibr 255(4):719–740
27. Yan JJ, Shyu KK, Lin JS (2005) Adaptive variable structure control for uncertain chaotic systems containing dead-zone nonlinearity. Chaos Solit Frac 25(2):347–355
28. Lin CM, Chen CH (2006) Adaptive RCMAC sliding mode control for uncertain nonlinear systems. Neural Comput Appl 15(1):253–267
29. Peng YF (2009) Robust intelligent sliding model control using recurrent cerebellar model articulation controller for uncertain nonlinear chaotic systems. Chaos Solit Fract 39(1):150–167